Project 6 Documentation | OOAD
Benjamin Kohav and Justin Astalos
Project Title: Fight Boat (BattleShip)

Work Done: Written description of the work done in the first week of your project and (in the case of multi-person teams) the breakdown of work across team members.

We jointly were able to create the initial JFrames, which Justin was able to start with the base files of JFrame, and then put together a first draft of how each page of the user interface would ultimately come together. From there, we worked on various aspects of the game, such as placing ships (Justin), and clicking on squares to "attack" on a board and change color (Ben). Ben also helped research how to make a clickable/interactable user interface for the grids/boards that we ultimately use to simulate two game boards for the user and the opponent.

Changes or Issues Encountered: Has anything changed so far in your approach to the project from the initial design in Project 5?

One issue/change we have encountered is figuring out the set up for our graphic user interface, where we would like to use JComponents on our JPanel to represent the two boards of the game (of player 1 and player 2), because they are clickable and simulate the users "attacking" a board spot using a click of the mouse. Understanding how to make a graphical user interface using JFrames has been very difficult to adjust to and learning what to use in order to properly create an interactive game where you can select places on a board to "attack" and "place" boats.

Change: We have had to reevaluate how we wanted to keep track of the game boards for player one and player two. We originally thought that we can store and present the game board of each player using JComponents, a feature of JPanel/JFrame, however we encountered many issues with "updating" the boards over the course of gameplay. So, we are now resorting to using a data

structure made up of a class called Square (which keeps track of a square in a game board), and we then use that data structure to create a Board of JComponents (with colors used to show what happened on each board game square).

Patterns: Now that you have more of your system implemented, please describe the use of design patterns so far in your prototype and how they are helping you or your design.

Singleton Pattern: Singleton is currently implemented in the code through the GameBoard class. It is a class which holds a single instance of itself that can be retrieved statically, and functions within it can update the instance. This makes the information about the game available for the user interface to access and update as needed, which makes use of Swing's JPanel more streamlined.

Plan for next iteration: Provide an estimate of how much more work needs to be done for your team to have implemented the design that you presented in Project 5 (with any design changes that may have occurred). What are your plans for the final iteration to get to the Project 7 delivery? What do you plan to have done when the overall project is due?

Currently, we estimate that our project is about halfway complete, because a large part of our plan outline in project 5 was to create this game specifically to be playable in a graphical user interface, which was not in the scope of this class and is a very new code style for both of us to be experimenting with. A lot of our time spent in project 6 was dedicated to understanding how we can apply our chosen Java package, JFrame, to actually make a user interface for our game. The first goal is just making the simplest possible implementation of the battleship game (start game, place boats, attack boards). The set up for our game has been more difficult than expected, but once it is complete, implementing the pattern designs and extra features that we outlined in Project 5. **Our plan for the next and final iteration** of the project is to first adjust our project

set up to properly keep track and store the game boards for each player in the simulation. Then, the main goal to start is to arrive at a simulation of the simple/basic Battleship game, and from there the design patterns we wanted to incorporate to make the game more interesting would be possible to implement. The design patterns in theory should not be too difficult to incorporate once we have a nice base version of the game, so we are confident that we will be able to include those in the final product.

Patterns:
Singleton
Observer
Strategy
Decorator

Red covers un-implemented features

Fight Boat Class Diagram

Justin Astalos, Ben Kohav | November 30, 2023